

Hierarchical Entity Path Notation Explained

When viewing or editing a transformation, you can see the instance code for each node or field as defined in the source or target structure to help you make an accurate mapping of fields. Depending upon your source and/or target data structure, instance codes consisting of elements, values, and attributes, along with any modifiers, are displayed next to each node or field name.

Elements, Values, and Attributes

The various [E] nodes are all represented by folders in the source or target mapping structures. The user maps to or from the [V], [A], or [EV] fields.

XML, JSON, NetSuite, Salesforce	Meaning
[E]	Element. In Jitterbit, elements are represented by nodes. Elements may have one or more other elements, attributes, values, a combination of these, or none depending on the modifiers.
[V]	Value. In Jitterbit, values are represented by fields. If a node has a value, it will also have at least one attribute. If it does not have an attribute, it will be represented as [EV].
[A]	Attribute. In Jitterbit, attributes are represented by fields. If a node has an attribute, it may or may not have a value field. There may be more than one attribute per node.
[EV]	Element value. In Jitterbit, these are elements that have a value but do not have any attributes and are thus represented by fields rather than nodes.



NOTE: The letter codes in the above table are not used within text or database structures.

Modifiers

Modifiers notate the relationship for nodes and sometimes fields as detailed in the table below.

XML, JSON, NetSuite, Salesforce	Complex text or database	Meaning
(no modifier)	Once Only	One and only one [1] element is valid. Only nodes can have this modifier.
+	One or More	One or more than one [1+] elements are valid. Only nodes can have this modifier.
?	Zero or One	Zero or one [0,1] element, value, attribute, or element value is valid. Both nodes and fields can have this modifier, which effectively means the node or field is optional.
*	Zero or More	Zero or more [0+] elements are valid. Only nodes can have this modifier.

Examples

When modifiers are used with elements, values, and attributes, consider the following examples:

Example 1

Assume there is an optional element called "Time." Then assume that this element has a value of type string, and does not have any attributes. Then it will appear as a field as follows:

```
[EV?] Time (String)
```

The user can then map directly from the [EV?] field.

Example 2

On This Page

- [Elements, Values, and Attributes](#)
- [Modifiers](#)
- [Examples](#)
- [Reference in a Script](#)

Related Articles

- [Converting a Database Source or Target to Use a JDBC Driver](#)
- [Data Normalization in Jitterbit](#)
- [Hierarchical Entity Path Notation Explained](#)
- [Mapping Flat to Hierarchical Output](#)
- [Mapping Flat to Hierarchical Structures](#)
- [Mapping Hierarchical to Flat Structures](#)
- [Multiple Mapping](#)
- [Source Loop Node](#)
- [Stream and Batch Transformations](#)
- [Transformation Request and Response Data Element Display](#)
- [Using a Database as a Target](#)

Related Topics

- [Transformations](#)

Last updated: Nov 16, 2017

If the "Time" element also has an attribute called "Timezone," then the element would be represented by a node, with value and attribute fields displayed within the node:

```
[E?] Time
    [V] (String)
    [A] Timezone (String)
```

In this case the [E?] node would appear as a folder in the tree, and the user would map to or from the [V] and [A] fields.

Reference in a Script

When referencing the elements, values, and/or attributes within a script, qualified paths are used. The qualified path is what appears within the <trans> </trans> tags in a Jitterbit script when referencing the specific node or field. The following symbols are used within the notation to signify the following:

Symbol	Meaning
[]	Square brackets are used to enclose nodes. However, note that node paths cannot be mapped directly; only fields can be mapped.
.	A period following a node name indicates a loop node.
\$	A dollar sign following a node name is used to indicate a <i>Once Only</i> or [E] element. This node cannot be a loop node.
#	A hash sign following a node name indicates that an array for the record value fields will be returned. This notation can be used only following a loop node before the period.
_	An underscore preceding <i>Root</i> is used to reference the root in a complex CSV. This is not present for other data structures. An underscore may also simply be an underscore already present in a node or field name.

The following sample paths are for a complex CSV data structure. The qualified paths for other structures follow the same format except that *_Root* is not present for other data structures.



NOTE: Not all possible combinations are shown, but can be constructed referencing the table above.

Data Structure	Qualified Path	Meaning
_Root (Once Only)	[_Root\$]	The <i>_Root</i> in a CSV data structure is always a node, and thus appears within brackets []. The ending dollar sign indicates a <i>Once Only</i> element. Node paths cannot be mapped.
Header (Once Only)	[_Root\$Header\$]	Sub-nodes use the reference of any preceding nodes, followed by the node name. The ending dollar sign indicates a <i>Once Only</i> or [E] element. Node paths cannot be mapped.
Value	[_Root\$Header\$Value]	A value field is not enclosed within square brackets, but still uses the reference of any preceding nodes, followed by the value name. This field path can be used for mapping.
Detail (One or More)	[_Root\$Header\$Detail.]	Sub-nodes use the reference of any preceding nodes, followed by the node name. The ending period indicates a loop node. Node paths cannot be mapped.
Attribute	[_Root\$Header\$Detail.Attribute]	An attribute field is not enclosed within square brackets, but still uses the reference of any preceding nodes, followed by the attribute name. This field path can be used for mapping.

<p>v a l u e [v]</p>	<p>_Root\$Header\$Detail#.Value</p>	<p>A value field is not enclosed within square brackets, but still uses the reference of any preceding nodes, followed by the value name. This specific example shows a hash sign preceding the loop node period, indicating that an array of the "Detail" record values will be returned. This field path can be used for mapping.</p>
--	-------------------------------------	---